



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁSTROJ PRO ROZŠÍŘOVÁNÍ OBRAZOVÝCH DATABÁZÍ

IMAGE DATABASE EXPANSION TOOL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Minařík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ilona Janáková, Ph.D.

BRNO 2020

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Jakub Minařík

ID: 203294

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Nástroj pro rozšiřování obrazových databází

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit nástroj pro rozšiřování počtu snímků databáze. Úkolem je navrhnout a implementovat různé techniky úprav reálných obrázků tak, aby vznikla širší a pestřejší databáze pro testování spolehlivosti metod počítačového vidění, případně jejich trénování. Uvažované úpravy jsou např. geometrické transformace (otočení, posunutí aj.), úpravy jasu a barvy a simulace některých vad (přidávání šumu, rozmazání, kapka, špína atd).

1. Seznamte se s danou problematikou. Proveďte rešerši existujících nástrojů a přístupů.
2. Zvolte vhodné testovací úlohy a pořídte si databáze reálných snímků.
3. Navrhněte vhodné techniky úprav snímků. Řešte i možnost jak změny lokalizovat a parametrizovat.
4. Vybrané techniky implementujte a otestujte na daných databázích.
5. Vytvořte vhodné uživatelské prostředí, které by umožňovalo nastavování vybraných úprav a dávkové spuštění na reálných snímcích.
6. Vše otestujte, stanovte omezující podmínky, zhodnoťte.

DOPORUČENÁ LITERATURA:

12. Computer Vision > 12.1. Image Augmentation. ZHANG, Aston, Zachary C. LIPTON, Mu LI a Alexander J. SMOLA. Dive into Deep Learning [online]. 2019 [cit. 2019-09-10]. Dostupné z: <http://www.d2l.ai>

Termín zadání: 3.2.2020

Termín odevzdání: 8.6.2020

Vedoucí práce: Ing. Ilona Janáková, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Tato práce se věnuje vytvoření nástroje pro rozšiřování obrazových databází. Práce se zabývá teorií zpracování obrazu a existujícím nástrojům a přístupům v této problematice. Z této teorie jsou navrženy jednotlivé metody, které jsou dále implementovány. Dále jsou tyto metody opatřeny uživatelským rozhraním a dávkovým spouštěním úprav obrázků. Nakonec jsou popsány omezující podmínky vytvořeného nástroje.

Klíčová slova

rozšiřování obrazových databází, augmentace dat, augmentace obrazu, zpracování obrazu, openCV, Qt

Abstract

This thesis devotes to create a tool for expansion of image dataset. Thesis deals with theory of image processing and existing tools and the approaches in this field. From this theory, individual methods are designed and implemented. Furthermore, these methods are provided by user interface and batch start of altering images. At the end restrictive conditions of program are described.

Keywords

expansion of image dataset, data augmentation, image augmentation, image processing, openCV, Qt

Bibliografická citace:

MINAŘÍK, Jakub. Nástroj pro rozšiřování obrazových databází. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/122769>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ilona Janáková.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Nástroj pro rozšiřování obrazových databází jsem vypracoval samostatně pod vedením vedoucí bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **8. června 2020**

.....
podpis autora

OBSAH

Obsah.....	4
Seznam obrázků.....	6
1 Úvod.....	7
2 Obraz, jeho zpracování a vlastnosti.....	8
2.1 Geometrické transformace.....	8
2.2 Jasové transformace.....	8
2.2.1 Transformace jasové stupnice.....	9
2.3 Barevné vlastnosti obrazu.....	10
2.4 Šum.....	11
2.4.1 Gaussův šum.....	11
2.4.2 Impulzní šum.....	11
2.4.3 Poissonův šum.....	12
2.5 Lokální filtry.....	12
2.5.1 Vyhlažování obrazu.....	12
2.5.2 Metody hledání hran.....	13
2.6 Obraz ve frekvenční oblasti.....	14
3 Existující nástroje, přístupy.....	15
3.1 Vybrané existující nástroje.....	15
3.1.1 Generativní adversariální sítě.....	15
3.1.2 Knihovny programovacích jazyků.....	15
3.1.3 Nástroje a programy.....	16
3.2 Augmentační metody.....	16
3.2.1 PatchShuffle transformace.....	16
3.2.2 Míchání obrázků.....	17
3.2.3 Náhodné vymazávání.....	17
4 Testovací metody a databáze.....	18
4.1.1 Testovací metody.....	18
5 Návrh Metod.....	19
5.1 Geometrické transformace.....	19
5.1.1 Zrcadlení.....	19
5.1.2 Rotace.....	19
5.1.3 Posunutí.....	20
5.1.4 Oříznutí.....	20
5.1.5 Škálování.....	20
5.1.6 Zkosení.....	20
5.2 Jasové a barevné transformace.....	20
5.2.1 Operace s charakteristikou jasu.....	21

5.2.2	Metody pracující s barvami.....	21
5.3	Přidání šumu	22
5.4	Lokální filtry	23
5.5	Vady objektivu	23
5.6	Lokalizace	24
6	Implementace vybraných metod	25
6.1	Popis fungování třídy a jejích částí AugMethods	25
6.1.1	Lokalizace	25
6.1.2	Nedefinované pixely	26
6.1.3	Implementace jednotlivých metod.....	26
7	Uživatelské rozhraní.....	30
7.1	Třída <i>CM2frame</i>	30
7.2	Třída <i>UndefPixFrame</i>	31
7.3	Třída <i>methodWidget</i>	31
7.3.1	Náhledový obrázek	32
7.3.2	Jednotlivé metody.....	33
7.4	Třída <i>MainWindow</i>	35
7.4.1	Načtení souborů	35
7.4.2	Zobrazení metod	35
7.4.3	Uložení.....	35
7.4.4	Spuštění generování obrázků	35
8	Zhodnocení a výsledky testů.....	37
8.1	Výsledky testů.....	37
8.2	Zhodnocení a omezující podmínky.....	37
9	Závěr.....	40
10	Literatura.....	41

SEZNAM OBRÁZKŮ

Obrázek 1 Geometrická transformace [1]	8
Obrázek 2 Ekvalizace histogramu: Vlevo před, vpravo po ekvalizaci [2]	9
Obrázek 3 Barevný model RGB a HSV [5] [6]	11
Obrázek 4 Impulzní šum	12
Obrázek 5 Použití Laplaceova filtru [8]	14
Obrázek 6 PatchShuffle [15]	16
Obrázek 7 Míchání obrázků: Kombinace tvaru a stylu [18]	17
Obrázek 8 Geometrické transformace: zleva - originál, zrcadlení, rotace	20
Obrázek 9 Redukce barev	21
Obrázek 10 Přičtení poloviny velikosti Hue	22
Obrázek 11 Vodní kapky – zaostřeno na kapky a velké kapky	29
Obrázek 12 Vodní kapky – bez kapek a malé kapky	29
Obrázek 13 Objekty třídy CM2frame	31
Obrázek 14 Třída UndefinedPixFrame	31
Obrázek 15 Třída methodWidget	32
Obrázek 16 Náhledový obrázek s lokalizací	33
Obrázek 17 Okno programu	36

1 ÚVOD

V dnešní době počítačové vidění zasahuje do mnoha oborů, kde slouží jako pomůcka a nástroj pro zjednodušení práce. Často je třeba snímky upravit a rozeznat objekty, které se na snímcích nachází. Pro tyto potřeby je užíváno mnoho metod, mezi ně patří i některé z oblasti umělé inteligence. Pro testování spolehlivosti a trénování těchto postupů jsou vyžadovány velké obrazové databáze. Rozšíření obrazových databází získáváme větší tréninkovou a testovací množinu. Metody rozšiřování obrazových databází pracují se změnou určitých informací z obrazu nebo jejím odebráním. To zamezení závislost klasifikačního modelu na různých informacích případně ověření jeho nezávislosti na daných informacích.

Cílem práce je vytvoření programu, který bude umožňovat rozšíření obrazových databází. Pro tento program budou navrženy metody, které budou vycházet z teorie o zpracování obrazu.

V první části bude sepsány možnosti zpracování obrazu a jeho vlastnosti. Budou popsána principy a použití základních úprav obrazu, jako jsou například geometrické transformace. Dále budou představeny vlastnosti obrazu, jako jsou například barvy a šum.

Existující nástroje a přístupy augmentaci obrazu budou popsány v druhé části. Také zde budou popsány existující metody přímo pro augmentaci obrazu, které zlepšují učení neuronových sítí.

V další části budou navrženy testovací metody pro mnou vytvořený program. Tyto testy by měli otestovat stabilitu a správnou funkčnost programu.

Při návrhu metod pro rozšíření obrazových databází bude použit teoretický základ z kapitoly zabývající se obrazem, tomu se bude věnovat další část.

V následující části se tyto metody implementují a ověří správnost návrhu, zdali je realizovatelný. Zde budou metody řešeny pouze pro spuštění na jednom obrázku.

Řešení dávkového spuštění bude probíhat následné části. Tato část se bude také věnovat řešení a návrhu uživatelského prostředí a spouštění celého programu.

Nakonec budou zhodnoceny výsledky testů. Popsány omezující podmínky a zhodnocení stavu programu.

2 OBRAZ, JEHO ZPRACOVÁNÍ A VLASTNOSTI

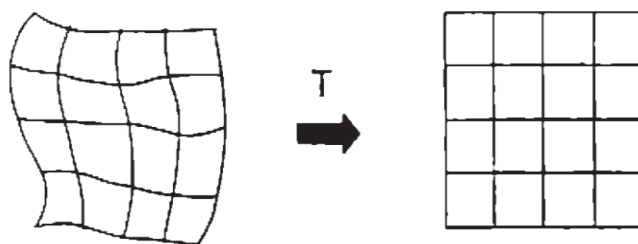
V této kapitole jsou popsány základní postupy zpracování obrazu. Mnoho postupů z této oblasti lze využít přímo nebo po úpravě k rozšíření obrazových databází. V této kapitole jsou popsány základní postupy zpracování obrazu a některé jeho vlastnosti a jejich možné úpravy. Popsány jsou geometrické transformace a jejich princip (jasové a barevné).

2.1 Geometrické transformace

Geometrické souřadnice počítají na základě původních souřadnic, nové souřadnice bodů. Ve zpracování obrazu se tyto transformace nejčastěji používají k odstranění nebo snížení vlivu geometrických zkreslení, jako jsou například geometrické vady objektivu. Geometrická transformace se dá rozdělit na dvě části z transformace souřadnic bodů, kde se vypočítávají nové souřadnice a aproximace jasové funkce, při které se zjišťuje nová jasová hodnota. [1]

Transformace souřadnic bodů

Transformace souřadnic bodů je krok, při kterém se ve vstupním obraze hledají odpovídající body ve výstupním obraze. Výstupní body pak mají spojitě souřadnice, protože poloha bodu nemusí souhlasit s celočíselnou mřížkou. Transformační vztah souřadnic bodů se obvykle aproximuje polynomem. V případě, kdy se geometrická transformace příliš rychle nemění, postačuje použití polynomu druhého a třetího stupně. [1]



Obrázek 1 Geometrická transformace [1]

Aproximace jasové funkce

Po zjištění nových souřadnic je třeba vypočítat novou jasovou hodnotu. Tato jasová hodnota by měla nejlépe odpovídat obrazu, který je výsledkem prvního kroku. Nejčastěji se pro zjištění nové hodnoty užívají jasové hodnoty v okolí nového bodu. Existuje více metod, jak spočítat novou hodnotu, pro názornost bude uvedeno pár příkladů.

Metoda nejbližšího souseda přiřadí hodnotu jasu nejbližšího bodu. Metoda lineární interpolace přiřadí hodnotu jasu dle čtyř bodů, tato hodnota odpovídá lineární kombinaci jasů těchto čtyř bodů. Vliv každého bodu je úměrný vzdálenosti. [1]

2.2 Jasové transformace

Jasové transformace se zpravidla provádí ve zpracování obrazu. Upravují obraz kvůli jasovým nedostatkům způsobeným nelinearitou citlivosti snímačů, nedostatečnému osvětlení, nečistotám na optické soustavě, případně pro zvýraznění některých prvků nebo zlepšení kontrastu pro snazší interpretaci člověkem. [1]

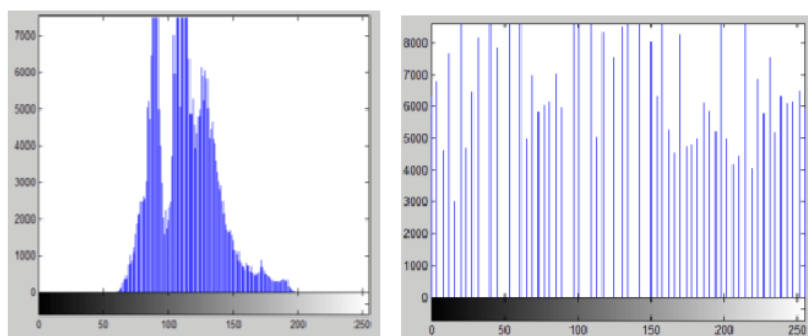
2.2.1 Transformace jasové stupnice

Transformace jasové stupnice je stejná pro všechny pixely v obraze. Jasové transformace se používají pro zviditelnění různých detailů v obraze pro lidské oko. Pro automatickou analýzu nemají takový význam a některé mohou díky možné ztrátě informací i uškodit.

Hodnoty jasu v šedotónovém mají typicky 256 úrovní, proto jsou jasové transformace poměrně jednoduše realizovány. Vytvoří se vyhledávací tabulka, která obsahuje nové jasové hodnoty. Podobný princip lze použít i pro barevné obrazy, každý kanál má svoji vyhledávací tabulku.

Typickými příklady jasové transformace jsou negativ nebo prahování a zvětšení kontrastu nebo ekvalizace histogramu.

Ekvalizace histogramu neboli vyrovnaní histogramu se velmi často používá pro zvýšení jasového kontrastu. Vyrovnaný histogram po transformaci jasové stupnice má zhruba stejně často zastoupeny jasové úrovně. [1] Ekvalizace lze použít i pro barevné obrazy.



Obrázek 2 Ekvalizace histogramu: Vlevo před, vpravo po ekvalizaci [2]

2.3 Barevné vlastnosti obrazu

Barvy v obraze je třeba pro potřeby digitálního světa číselně reprezentovat. Proto se využívají barevné modely a prostory. Barevný model je matematický model, který převede barvy na číselnou reprezentaci. Barevný prostor je implementovaný barevný model, jak mají být reprezentovány jednotlivé složky modelu. V další části budou představeny některé běžně užívané modely. [3]

2.3.1.1 RGB

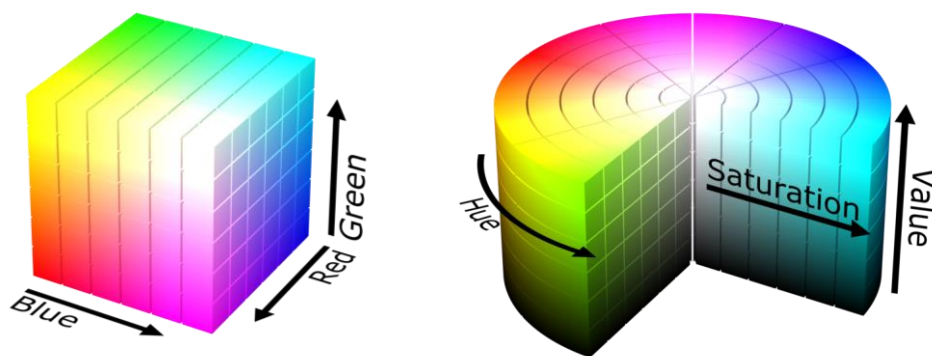
V počítačové technice nejpoužívanějším barevným modelem je RGB. Tento barevný model tvoří barvy sčítáním tří základních barev – červené, zelené a modré. Tyto barvy nabývají hodnot v intervalu $\langle 0,1 \rangle$, pro lepší reprezentaci je mnohem známější v počtu bitů, tedy je-li barva uložena v 8 bitech nabývá 256 hodnot. Tento model využívají digitální kamery a obrazovky. Model je možné zobrazit jako krychli v souřadnicovém systému (Obrázek 2). [4]

2.3.1.2 CMY

CMY je barevný model podobný modelu RGB, avšak jedná se o model subtraktivní, barvy se odečítají. Jestliže jsou zastoupeny všechny složky dostaneme černou barvu. Barvy, které model používá, jsou azurová (cyan), purpurová (magenta) a žlutá (yellow). Tento model je nejčastěji používán v tiskárnách.

2.3.1.3 HSV

Na rozdíl od předchozích tento model nevyužívá pro reprezentaci jednotlivé barvy. Proto je také tento model mnohem bližší popisu člověka. Barva je zde také zastoupena třemi složkami. Ty jsou barevný odstín (hue), sytost (saturation) a jas (value). Barevný model bývá zobrazován jako kužel nebo válec. Kdy barevný odstín nabývá hodnot v intervalu $\langle 0^\circ, 360^\circ \rangle$, určuje, jaká barva je použita. Sytost se udává v intervalu $\langle 0,1 \rangle$ a představuje vodorovnou osu, na které určuje, jakým podílem je barva zastoupena. Jas je také udáván v intervalu $\langle 0,1 \rangle$ a představuje svislou osu a udává množství bílého světla. Model bývá znázorněn pomocí válce nebo kužele, kdy vrchol kužele je na spodní straně (obrázek 2). [4]



Obrázek 3 Barevný model RGB a HSV [5] [6]

2.4 Šum

Šum, jako takový, je přítomný v každém obraze. Může být způsoben technikou, která snímek pořizovala; podmínkami, při kterých byl pořízen; špatným zpracováním, přenosem obrazu nebo i jejich kombinacemi. Existuje mnoho typů šumu a mnoho typů dělení. Může být tříděn podle toho, jak ovlivňuje hodnotu, jestli se k ní přičítá, násobí ji; zdali ovlivňuje celou oblast obrazu nebo jen některé pixely. [7]

2.4.1 Gaussův šum

Gaussův šum je měřitelný šum. Má funkci hustoty pravděpodobnosti, která je stejná jako funkce normálního rozložení neboli Gaussova rozložení. Projevuje se přičtením náhodné hodnoty z normálního rozložení ke každému pixelu celého obrazu. Zdrojem bývá šum senzoru, nevhodné osvětlení, elektrický obvod a tepelný šum. [7]

2.4.2 Impulzní šum

Impulzní šum je často také nazývaný jako šum „Pepř a sůl“. Výsledkem tohoto šumu jsou ostré a náhodné změny v obraze. Tento šum nezasahuje do celého obrazu, ale jen některá místa. Postižené místa tímto šumem nabývají extrémních hodnot, tyto hodnoty jsou ve většině případů černé nebo bílé barvy. Šum nejčastěji vzniká chybami jednotlivých buněk senzoru, paměťových buněk nebo chybami při digitalizaci a přenosu obrazu. [7]



Obrázek 4 Impulzní šum

2.4.3 Poissonův šum

Poissonův šum neboli fotonový šum vzniká ve snímačích, které využívají malé množství částic, energie a malý náboj. Jeho výskyt je popsán Poissonovým rozdělením pravděpodobnosti. Jde o aditivní šum, který je závislý na jasové hodnotě pixelu. [7]

2.5 Lokální filtry

Lokální filtrace je konvoluce dvou signálů, filtru a obrazu. Při operacích s diskrétním obrazem se tato operace značně zjednoduší. Při použití lokálního filtru je hodnota daného pixelu vypočítána z hodnot blízkého okolí. Je to oblíbená technika k rozmazání a detekci hran. Metody lokální filtrace se dají rozdělit na dvě skupiny vyhlazování a detekci hran. Vyhlazování má za účel potlačit vlivy šumu v obraze. Při detekci hran zjišťujeme derivaci funkce, která zachycuje extrémy, tedy hrany v obraze. Jedno z dalších možných rozdělení je na lineární a nelineární filtry. Využívá se lokální konvoluce jádra, které mu se říká konvoluční maska. Většinou se používá čtvercové okolí s lichou velikostí strany, kdy nový bod, který je počítám, pak leží ve středu. [1]

2.5.1 Vyhlazování obrazu

Metody vyhlazování obrazu pomocí lokálních filtrů mají za úkol snížit vliv šumu na obraz. Způsoby, jakými se vypočítá nová hodnota, se dají rozdělit na lineární a nelineární.

Používají se tzv. prostorově invariantní filtry, což znamená, že jejich chování se při změně polohy v obraze nemění. Skutečný obraz tento předpoklad neumožňuje, protože jeho hodnoty nemůžou být záporné a jsou omezené. Proto platí prostorová invariantnost filtrů jen pro některé posuny konvoluční masky. To

se většinou řeší použitím lineárního přístupu a přidáním dodatečných podmínek pro okraje obrazu a meze rozsahu hodnoty.

Obyčejné průměrování je základní lineární metodou vyhlazování obrazu. Každému bodu je přiřazena hodnota, která je aritmetickým průměrem ostatních. V některých případech se konvoluční maska změní zvětšením váhy středu nebo jeho čtyř sousedů nebo použitím Gaussova rozložení. Nevýhodou těchto metod je rozmazání hran. Následující vztahy ukazují tyto masky pro okolí 3x3:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.1)$$

Rozmazání hran je částečně řešeno nelineárními filtračními postupy. Jedním z nejjednodušších je filtrování pomocí mediánu, kdy z daného okolí je jako nová hodnota vybrán medián daných hodnot.

Dalším zástupcem je metoda rotující masky. Kolem bodu rotuje maska a vzniká několik okolí. Výběr okolí je dán podmínkou homogenity, například nejmenším rozptylem. Nová hodnota může být vypočtena průměrem hodnot. Metoda nerozmazává hrany obrazu, ale mírně je ostří. Nevýhodou je časová náročnost výpočtu nového obrazu. [1]

2.5.2 Metody hledání hran

Matematickým nástrojem, který je používám k hledání hran, je gradient. Pixely, které mají velkou hodnotu gradientu, se nazývají hrany. Pro hledání hran se využívají také konvoluční masky. Existují směrové filtry, které obsahují tolik masek, kolik směrů rozlišují. Velká nevýhoda těchto filtrů je velké ovlivnění šumem. Filtry se mohou využít i pro ostření hran v obrazu. To se provádí odečtením vyfiltrovaného obrazu od původního.

Laplaceův operátor je operátorem druhé derivace, proto je invariantní vůči otočení a udává jen velikost hrany. Nevýhodou je velká citlivost na šum a dvojité odezvy na hrany odpovídající tenkým liniím. [1] Příklad používaných konvolučních masek pro Laplaceův operátor:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.2)$$



Obrázek 5 Použití Laplaceova filtru [8]

2.6 Obraz ve frekvenční oblasti

Pro převedení obrazu do frekvenční oblasti se používá Fourierova transformace. Frekvenční oblast umožňuje jednoduše filtrovat určité frekvence. Používá se pro odstranění nechtěného opakujícího se vzoru, kdy je frekvence daného vzoru odstraněna.

Operace konvoluce je ve frekvenční oblasti pouhé násobení těchto signálů. To znamená ušetření výkonu pro větší filtry nebo při použití více operací. Pro malé filtry je naopak provedení konvoluce ve frekvenční oblasti náročnější, protože převod do frekvenční oblasti a zpět se zde znatelně projevuje.

3 EXISTUJÍCÍ NÁSTROJE, PŘÍSTUPY

Existuje mnoho nástrojů umožňujících augmentaci dat, zvláště pak dat obrazových. Většina z nich jsou udělány jako programy pro příkazové řádky, nebo přímo jako knihovny, které je možné přidat do programu (nejčastěji jsou dělány v jazyku Python). Stále jsou vyvíjeny a testovány nové metody rozšiřování obrazových databází, které by zlepšily kvalifikaci objektů nebo rychlost učení neuronových sítí. Nejčastěji jsou tyto nástroje a přístupy vytvořeny pro zlepšení výsledků neuronových sítí.

3.1 Vybrané existující nástroje

V následujících podkapitolách bude uvedeno pár příkladu používaných nástrojů.

3.1.1 Generativní adversariální sítě

Generativní adversariální síť je neuronová síť, která ke svému učení využívá augmentaci dat. Jedná se o dvě neuronové sítě, které spolu soupeří. Jedna síť, generativní, tvoří nová data na základě dodaných a druhá ty data hodnotí podle toho, jak moc jsou podobná vstupním datům. [9] Tento proces lze použít pro generování nových dat. [10]

3.1.2 Knihovny programovacích jazyků

Velké množství vytvořených nástrojů na augmentaci dat jsou knihovny do programovacích jazyků, nejčastěji do Pythonu. Dva nejpoužívanější z nich zde budou popsány.

Albumentations

Knihovna do jazyka Python, postavená na knihovnách a balíčcích numpy, OpenCV a imgaug. Obsahuje více než 30 augmentačních metod. [11]

imgaug

Další velmi často používaná knihovna, také do jazyka Python, Autoři ji popisují: „Vytvoří ze souboru obrázků, nový mnohem větší soubor lehce změněných obrázků.“ [12]

3.1.3 Nástroje a programy

Další z možností, jak augmentaci dat řešit, je pomocí nástrojů, které tyto vlastnosti řeší. Většina z nich funguje formou programu v konzoli nebo skriptu, kdy se při spuštění předají potřebné parametry a data.

UDA – Unsupervised data augmentation

Jednoduchý augmentační nástroj zamýšlený pro práci se strojovým učením. Nástroj projde složku s obrázky a vygeneruje nové na základě specifikovaných úprav pro každý obrázek který najde. [13]

Data Augmentation For Object Detection

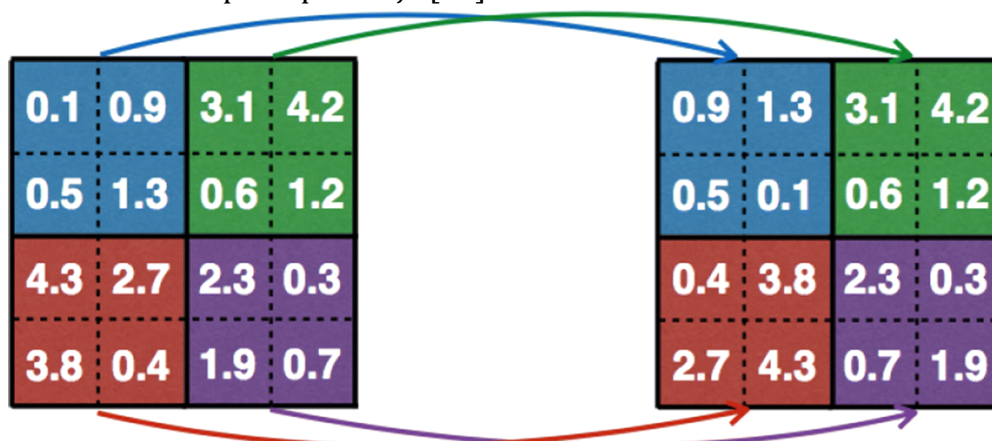
Tento nástroj se zaměřuje na rozpoznávání objektů a umožňuje provádět změny na obraze, a zároveň tyto změny provede i na hranicích, které označují daný objekt. [14]

3.2 Augmentační metody

Většina metod a postupů, které se využívají k rozšiřování obrazových databází, jsou převzaté z jiných oblastí počítačového vidění, ale existují a vznikají nové metody, které jsou vyvíjeny přímo pro augmentaci obrazů.

3.2.1 PatchShuffle transformace

Vyhlazování a hledání hran jsou klasické způsoby použití filtračních jader. Tato metoda ve vybraném prostoru náhodně prohodí pixely. Názorná ukázka je na obrázku 5, kde je znázorněna matice 4x4 se čtyřmi prostory, pixely v těchto prostorech se náhodně přeuspořádají. [15]



Obrázek 6 PatchShuffle [15]

3.2.2 Míchání obrázků

První metoda, která kombinuje dva obrázky, byla navržena pro trénování neuronových sítí. Náhodným výběrem jsou vybrány dva obrázky z databáze nehledě na to, zdali znázorňují stejný objekt. Tyto obrázky se poté zkombinují dohromady, průměrováním hodnot pixelu. Přestože výsledné obrázky pro člověka nedávají smysl, při správném užití mohou zlepšit naučení neuronových sítí. [16] Tato metoda byla později rozšířena i o různé nelineární způsoby spojení dvou obrázků dohromady. [17]

Další metoda kombinující dva obrázky převádí styl jednoho na druhý. Což znamená, že na tvar objektu na jednom obrázku nanese vzor a texturu druhého, ukázka na obrázku 6. [18]



Obrázek 7 Míchání obrázků: Kombinace tvaru a stylu [18]

3.2.3 Náhodné vymazávání

Další metodou je metoda náhodného vymazávání, taktéž pro učení neuronových sítí. Tato technika byla navržena pro zlepšení výsledků při situacích, kdy některé části objektu byly nejasné. Metoda vytvoří obdélník, který je vyplněn různými způsoby, např. bílou barvou. Tento obdélník je pak náhodně umístěn do obrazu. [19]

4 TESTOVACÍ METODY A DATABÁZE

Testovací metody budou prováděny na vytvořených databázích. Tyto databáze tvoří fotky různých předmětů, kdy každý předmět je nafocen z jiných úhlů, natočení, vzdálenosti a rozdílných světelných podmínek.

4.1.1 Testovací metody

V této části budou popsány jednotlivé testovací metody, které ověří správnou funkci vytvořeného programu. Všechny testovací metody budou provedeny pro každou vytvořenou databázi.

4.1.1.1 Vytvoření maximálního počtu prvků

Každá metoda bude otestována vytvořením maximálního počtu snímků. To bude otestováno na všech pořízených databázích. Pokud nebude určena největší maximální hranice, bude to vyzkoušeno pro velmi velké číslo tak, aby nedošlo k přespřílišnému přetížení počítače, na kterém se program testuje. Možné chyby budou přetečení číslování pro názvy ukládaných obrázků, neprovádění změn od určitého počtu snímků, neukládání výsledných obrázků u velkých počtů, případně další různé chyby.

4.1.1.2 Vyzkoušení všech možností parametrizace a lokalizace

Vyzkoušením všech možností parametrizace a lokalizace všech metod se dosáhne otestování správného předávání parametrů z uživatelského prostředí do funkcí upravující obraz. Ověří se spuštění funkcí pro všechny parametry.

4.1.1.3 Ověření výběru obrázků

Tento test bude ověřovat výběr všech obrázků při stejném počtu vstupních a výstupních obrázků, dále náhodnost výběru pro větší a menší počty obrázků, například zdali obrázky nejsou brány popořadě, nebo zdali není vybrán vždy pouze první.

4.1.1.4 Ověření náhodných možností parametrizace

Vyzkoušení výběru náhodných parametrů může objevit chybu implementace, která by výrazně změnila pravděpodobnost výběru parametru, například zdali výběr nekonverguje k některé z krajních hodnot.

4.1.1.5 Správnost uložení a načtení souborů

Test má za úkol vyzkoušet správné načtení všech podporovaných formátů a jejich správné uložení ve správné lokaci.

5 NÁVRH METOD

V této kapitole budou popsány vybrané metody pro rozšíření obrazových databází. Soupis těchto metod vychází z literatury a předešlé kapitoly. I přes užívání a existenci různých metod, ty nejpoužívanější jsou velmi často ty stejné, preferované pro svoji jednoduchost a časovou a výpočetní nenáročnost.

V mnoha aplikacích lze použít jen některé metody. Některé obrazové databáze vyžadují, aby dané vlastnosti a prvky zůstaly nezměněny. Například u snímků mozku, které jsou centrovány a pořízeny ve stejné pozici, je nežádoucí, aby byl snímek posunut nebo rotován. Podobné podmínky platí nejen u metod pracujících s polohou, ale i s barvou a dalšími vlastnostmi. [20]

5.1 Geometrické transformace

Geometrické transformace snižují závislost modelu na pozici. Geometrické transformace jsou pro svoji jednoduchost a rychlou implementaci velmi používané pro rozšiřování obrazových databází. Nevýhodou pak může být paměťová náročnost a vyšší výpočetní náročnost. Některé z metod musí být kontrolovány, zdali nezasáhly přes příliš do obrazu, např. oříznutí. [20]

U některých geometrických transformací vznikají prázdná místa, pixely, které nemají jednoznačnou hodnotu a do obrázku byli přidány pro zachování formátu. Tyto prázdné oblasti se mohou vyplnit jednotnou barvou nebo barvou nejbližšího pixelu nebo původní hodnotu.

5.1.1 Zrcadlení

Je velmi jednoduchá, a i přesto efektivní metoda. Patří mezi první a nejrozsáhleji užívané metody. Nově vygenerovaný obrázek má stejné rozměry jako původní. Jedná se o osovou souměrnost, podle jedné strany obrazu. Lze jej aplikovat v jedné nebo v obou osách zároveň. Tato metoda není vhodná pro všechna užití, například při snímcích objektu, jehož určité části vždy v určité oblasti snímku nelze použít, protože by tyto části mohly změnit svoji polohu. [21] [22]

5.1.2 Rotace

Rotace obrázek otočí podle daného středu o určitý úhel, kdy je obrázek celý otočen. Při rotaci může část obrazu přesahovat, přesahující pixely jsou zahozeny a vznikají místa, která nemají danou hodnotu. [21] [22]



Obrázek 8 Geometrické transformace: zleva - originál, zrcadlení, rotace

5.1.3 Posunutí

Metoda posunutí posune obrázek o určitý počet pixelů v jedné nebo obou osách, vzniká pak prázdné místo nedefinovaných pixelů. Užití metody posunutí je vhodné, pokud chceme docílit menší závislosti modelu na poloze předmětu v obrázku.

5.1.4 Oříznutí

Část obrázku je oříznuta, může být ořezán ve všech osách. Výsledný obrázek může mít velikost oříznuté oblasti nebo obsahovat prázdné místo s nedefinovanými pixely. U těchto pixelů nemá smysl nechávat jejich původní hodnotu, jelikož by operace ztratila smysl a vyšel by nám počáteční obraz.

5.1.5 Škálování

Metoda zahrnuje zmenšení nebo zvětšení obrazu dle koeficientu, který násobí délku dané osy, což znamená změnu rozlišení obrazu. Obraz je možná ponechat v nově vzniklé velikosti nebo jej oříznout případně vyplnit na velikost původní.

5.1.6 Zkosení

Obrázek je zkosen dle daného koeficientu v dané ose. Při této operaci vznikají zase prázdná místa.

5.2 Jasové a barevné transformace

Jasové transformace pracují s jasnem, buď mu přiřazují novou hodnotu nebo jej upravují. Pracují buď pouze s danou hodnotou pixelu na základě společného pravidla a charakteristiky. Jasové transformace mohou být prováděny i na barevných kanálech. Barevné transformace a metody pracují s více jasovými kanály, to jim naskýtá více možností, jak obraz upravit.

5.2.1 Operace s charakteristikou jasu

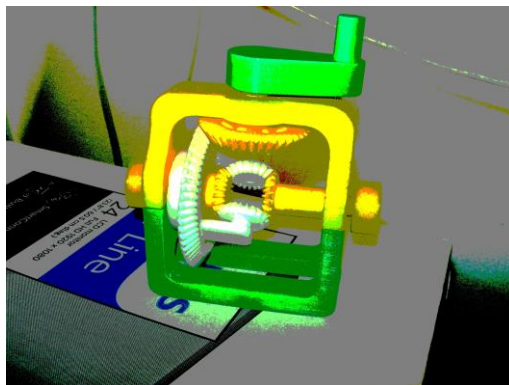
Velké množství metod jasových a barevných transformací se dá znázornit převodní charakteristikou jasu mezi vstupním a výstupním obrazem.

5.2.1.1 Prahování

Prahování převádí jas na omezený počet úrovní, obvykle dvě. Zlom mezi dvěma úrovněmi se nazývá práh prahování. Použití pro augmentaci obrazových databází má pravděpodobně větší význam při více úroňovém prahování. Dvou úroňové prahování by mohlo pro většinu příkladů odebrat z obrazu informací přespříliš.

5.2.1.2 Redukce barev

Redukce barev funguje na stejném principu jako prahování, daným číslem se je vydělen počet barev na výsledný, pro barevný obraz je minimální počet barev 6.



Obrázek 9 Redukce barev

5.2.1.3 Úprava kontrastu a jasu

Úprava jasu přičítá nebo odčítá od všech jasových hodnot dané číslo. Při změně kontrastu jsou všechny hodnoty jasu vynásobeny danou změnou kontrastu.

5.2.1.4 Inverze

Inverzí dojde k převrácení všech jasových hodnot, jestliže jsou hodnoty od 0 do 255 a hodnota je nula tak po inverzi je rovna 255.

5.2.2 Metody pracující s barvami

Dalším prostředkem pro úpravu obrazu pro augmentaci je barevná úprava. Tyto metody docílí nezávislost modelu na barvách. Při použití je třeba myslet, zdali je možné využít metody, které upravují barvy a jak je potřeba aby byla databáze závislá na barvách. [3] Metody mohou být použity v různých barevných prostorech s jinými výsledky.

5.2.2.1 Zamíchání kanálů

Zamíchání barevných kanálů prohází jejich barevné zobrazení, například z červené se stane zelená a naopak, pořadí může být buď určeno nebo náhodné.

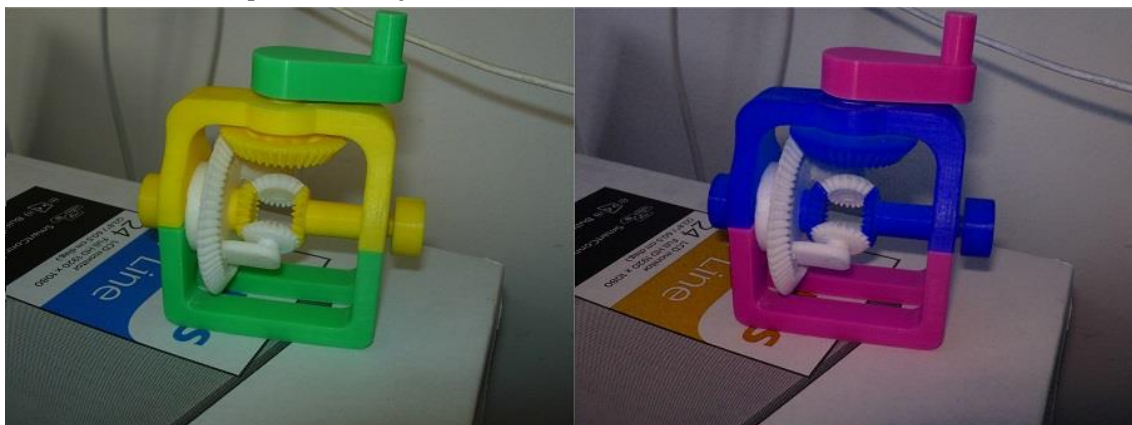
5.2.2.2 Převedení do šedotónového obrazu

Obraz může být převeden do šedotónového obrazu buď průměrováním všech hodnot nebo užitím vztahu, který určuje poměr podle citlivosti lidského oka.

5.2.2.3 Přičtení hodnoty k barevnému kanálu

Při této metodě je konstanta přičtena k barevnému kanálu. Ke všem kanálům mohou být přičteny nebo odečteny různé konstanty, při přerostení hodnoty tak hodnota zůstává na maximální, respektive minimální hodnotě. Může být použita v HSV a RGB barevném prostoru.

Při přičítání v barevném prostoru HSV, je kanál H jsou hodnoty udávány ve stupních, po dosažení maximální hranice se hodnoty zapisují zase od začátku. Barevný odstín se tak může různě posouvat po celé kružnici. Při přičtení pouze ke kanálu H se mění pouze barvy.



Obrázek 10 Přičtení poloviny velikosti Hue

5.2.2.4 Nastavení celého kanálu na konstantní hodnotu

Tato metoda nastaví kanál, ať už v prostoru HSV nebo RGB na libovolnou konstantní hodnotu.

5.3 Přidání šumu

Přidání šumu může měnit pouze některé pixely nebo celý obraz, podle zvoleného typu šumu.

5.3.1.1 Impulzní šum

Impulzní šum neboli šum typu pepř a sůl, mění náhodné pixely buď na černou nebo bílou barvu. Kolik pixelů je ovlivněno, může být určeno procentem z celkového počtu pixelů.

5.3.1.2 Gaussův šum

Gaussův šum se do obrázku přidává aditivně. Ovlivňuje tedy všechny pixely. Velikost ovlivnění toho to šumu se upravuje parametry normálního rozdělení, tedy střední hodnotou a rozptylem.

5.4 Lokální filtry

Užití lokálních filtrů pro vyhlazování obrazu, případně rozmazání má za následek zlepšení rozpoznávání rozmazaného objektu v pohybu. Naproti tomu zaostření obrazu může mít za následek ponechání více detailů o objektu. Nevýhodou těchto metod je podobný princip fungování jako u vnitřních mechanismů konvolučních neuronových sítí. Podobné výsledky proto mohou být dosaženy úpravou parametrů standartní konvoluční vrstvy.

5.4.1.1 Rozmazání

Rozmazání je prováděno průměrovacím filtrem s možností určení velikosti daného filtru.

5.4.1.2 Gaussův filtr

Gaussův filtr rozmazává pomocí filtru vytvořeného z Gaussovy křivky, je možné určit velikost a rozptyl.

5.4.1.3 Medián

Filtr pomocí mediánu vybírá z dané čtvercové oblasti, jejíž velikost je libovolná, střední hodnotu.

5.4.1.4 Vlastní filtr

Vlastní filtr, umožní zvolit si libovolný filtr, pravděpodobně s omezenou velikostí, s možností násobit celý filtr konstantou.

5.5 Vady objektivu

Dalšími možnými metodami rozšíření databází je simulace vad objektivu a jeho znečištění.

5.5.1.1 Zkreslení objektivu

Tato metoda umožní simulovat geometrické zkreslení objektivu. Geometrické zkreslení radiální, pro soudek a podušku, a tangenciální bude parametrizováno podle rovnic: [26]

$$x' = x \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (5.1)$$

$$y' = y \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (5.2)$$

$$x' = x + 2p_1 y + p_2 (r^2 + 2x^2) \quad (5.3)$$

$$y' = y + 2p_1 x + p_2 (r^2 + 2y^2) \quad (5.4)$$

Kde x a y jsou původní souřadnice pixelu, x' a y' jsou nové souřadnice daného pixelu, r je poloměr od středu obrazu, k_1, k_2, k_3 jsou koeficienty radiálního zkreslení a p_1, p_2 jsou koeficienty tangenciálního zkreslení. Tyto rovnice je možné sečíst a aplikovat obě zkreslení zároveň.

5.5.1.2 Vodní kapky

Tato metoda má za úkol simulovat kapky vody na objektivu. Ke své funkci by měla využít rozdílů mezi skutečnými fotkami, s kapkami a bez kapek a ty to zjištěné rozdíly aplikovat na obrázek.

5.6 Lokalizace

Lokalizace metod bude probíhat pro všechny metody stejně. Její fungování bude fungovat jako vyříznutí obdélníkové oblasti z daného obrazu. Tato oblast pak bude upravena a vložena zpět do původního obrazu. Tento princip umožňuje jednoduše lokalizovat a omezit oblast obrazu, která má být pozměněna. Tuto oblast bude zadávat uživatel.

6 IMPLEMENTACE VYBRANÝCH METOD

Pro implementaci byl zvolen jazyk C++. Pro lepší práci s obrazem byla použita knihovna OpenCV, její výhoda je v obsáhlosti, všestrannosti v rámci počítačového vidění a značná popularita, což umožňuje rychlejší řešení případných problémů a nejasností v rámci dokumentace.

Mnoho metod a funkcí byla řešena pomocí funkcí zabudovaných funkcí z OpenCV, které jsou velmi dobře optimalizovány. Podporované K návrhu funkcí byla použita dokumentace OpenCV. [27]

6.1 Popis fungování třídy a jejích částí AugMethods

Metody pro rozšíření obrazových databází jsou implementovány pomocí třídy AugMethods. Třída je vytvořena aby se při jejím použití nebylo třeba užívat proměnných a funkcí knihovny OpenCV. Snahou při psaní této třídy bylo oddělit OpenCV funkce od funkcí grafického rozhraní.

Po zvážení předávání parametrů funkcím a pro více možností začlenění do uživatelského rozhraní, byly funkce udělány pro zpracování jednoho obrázku, tedy není v dané třídě začleněna smyčka, která by byla přímo v metodách nebo je aplikovala. To znemožňuje použití bez grafického rozhraní i po drobných úpravách. Naopak tato možnost umožní při návrhu grafického prostředí nevázat se na přesně dané předávání parametrů do smyčky.

Při tvoření třídy bylo snahou zmenšit počet inicializací paměti. Jedna z možností byla udělat vnitřní proměnné statické a při volání konstruktoru třídy již nahrát obrázek. To by znamenalo pro každý nový obrázek inicializovat novou instanci třídy. Po promyšlení bylo tohle rozhodnutí změněno a tvoří se pouze jedna instance a pouze se mění nahraná data v ní.

Po inicializaci instance se pro většinu metod zadá adresa obrázku a ten je nahrán do vnitřních proměnných instance, ty jsou čtyři. Dvě uchovávají původní obrázek a výsledný. Zbývající dvě jsou pro práci s obrazem. V dané třídě jsou ještě další pomocné proměnné.

V třídě je také zahrnuta funkce, která překopíruje vnitřní proměnné tak aby se mohla libovolná metoda užít znovu na již alternovaný obraz.

Třída umožňuje, pro realizaci náhledu, převedení některých vnitřních proměnných typu *Mat* z OpenCV do typu *QImage* pro knihovnu Qt.

6.1.1 Lokalizace

Lokalizace se určuje ještě před nahráním obrázku to umožňuje, dosadit parametry lokalizace před smyčkou ve které se budou obrázky zpracovávat.

Parametry lokalizace jsou uloženy ve vnitřní proměnné procentuálně a při nahrání obrázku jsou převedeny a přepočítány dle velikosti daného obrázku.

6.1.2 Nedefinované pixely

Při některých geometrických transformacích vznikají oblasti, které nemají definovanou hodnotu. Pro dané pixely je možnost vybrat konstantní barvu nebo původní hodnotu pixelů. Funkce, které to umožňují mají dané nastavení na transparentní, pro dané pixely, tedy jsou brány pixely v místě kam se má obraz uložit. Při nahrání obrázku je obrázek zkopírován do proměnných, kam se ukládá z daných funkcí výsledek. To má za následek defaultní nastavení těchto pixelů na hodnotu původní. Pro změnu pixelů na konstantní hodnotu je implementována funkce, která nastaví dané proměnné na danou barvu.

6.1.3 Implementace jednotlivých metod

Mnoho metod bylo implementováno pomocí vestavěných funkcí OpenCV, jednak zaručují spolehlivost, ale i značnou optimalizaci daného procesu. Tyto metody budou v následujícím popisu vynechány. Pro všechny funkce platí, že výstupní obrázek je stejně velký jako vstupní.

6.1.3.1 Rotace

Funkce rotace má možnost otočit obraz pro libovolný úhel. To se provádí pomocí afinní transformace, kterou zajišťuje funkce z OpenCV. Pro zrychlení je využito funkce, která rotuje obrázek o násobky 90°, ta je přidána pouze pro úhel 180°. Pro úhly 90° a 270° nebyla přidána z důvodu zachování stejné velikosti výstupu jako vstupu.

6.1.3.2 Posunutí

Posunutí je implementované pomocí výběru části obrazu. Tato část je pak hloubkově zkopírována do výběru cílové proměnné. Před kopírováním je obsažena logika, která přepočítá posunutí v obou osách na cílové oblasti. Parametry je možné zadávat v procentech nebo pixelech.

6.1.3.3 Oříznutí

Oříznutí je uděláno pomocí výběru obdélníkové oblasti dvou bodů. Daná oblast je poté zkopírována do cílové proměnné, pokud nebyli nedefinované pixely nedefinovány na konstantní barvu, k oříznutí nedojde a výsledkem bude stejný obrázek.

6.1.3.4 Škálování

Tato funkce umožňuje zvětšení a zmenšení obrázku. Podle toho zdali se obrázek zvětšuje nebo zmenšuje je vybrán způsob aproximace jasových hodnot. Po volání funkce pro změnu velikosti z OpenCV, která obrázek zvětší nebo zmenší je počítána oblast, která se přiřazuje do cílové proměnné, aby obrázek neměnil velikost.

6.1.3.5 Metody pracující s vyhledávací tabulkou

Většina operací pracující s barvami a jasnem byla realizována pomocí vyhledávací tabulky, to umožňuje snížit časovou náročnost a definování pouze jedné tabulky pro více obrázků. Tabulku je možné vytvořit před nahráním obrázku. Poté co je tabulka vytvořena je uložena ve vnitřní proměnné.

Funkce pro vytvoření tabulky vytváří tabulku se třemi kanály. Každý z těchto kanálů je na sobě nezávislý. Funkce vytváří tabulku pro negativ, prahování, změnu kontrastu a jasu nebo ponechání daného kanálu beze změny.

Při volbě prahování je možné vybrat libovolný počet prahů a určit hodnoty mezi nimi. Možnost změny kontrastu a jasu upravuje hodnoty v tabulce jako přímku.

Vyhledávací tabulku vytváří i funkce pro redukci barev. Ta vypočítá počet prahů a hodnoty mezi nimi a data použije pro vytvoření vyhledávací tabulky pomocí předchozí funkce.

Pro aplikování vyhledávací tabulky se spouští funkce z OpenCV, která aplikuje danou tabulku na obrázek.

6.1.3.6 Přičtení nebo nastavení hodnoty daného kanálu

Kanály daného obrazu jsou rozděleny, aby k nim byl umožněn přístup po jednom. Přičtení a odečtení hodnoty od celého kanálu je jednoduchá operace využívající operátoru plus. Nastavení konstantní hodnoty využívá operátoru rovná se.

Pro přičtení k hodnotě kanálu H z HSV, byla vytvořena funkce. Tato funkce zajišťuje, že hodnoty, které přetečou budou přiřazeny zase od nuly. Toho je dosáhnu to využitím operátoru modulo.

6.1.3.7 Impulzní šum

Vstupním parametrem funkce je procentuální počet ovlivněných pixelů oproti velikosti obrázku a barva pixelů, zdali bílá černá nebo jejich kombinace. Ve funkci není ošetřeno, aby se náhodně vybrané pixely nepřekrývali, proto celkový počet pixelů je menší nebo roven danému parametru změněných pixelů.

Funkce je řešena dvěma smyčkami vnitřní prochází dané náhodné pixely a přiřazuje jim danou barvu. Druhá smyčka umožňuje změnu barvy vybraných pixelů na černou, bílou nebo jejich kombinaci. Pokud je vybrána kombinace barev je druhá

smyčka provedena dvakrát a pro každé provedení je jiná barva. Následkem toho je počet cyklů vnitřní smyčky snížen o polovinu, aby mohla změnit se mohla změnit barva.

6.1.3.8 Gaussův šum

Pro vytvoření Gaussova šumu je využita funkce *randn* z OpenCV, jako vstup této funkce je střední hodnota a směrodatná odchylka, která je vypočítána ze vstupního rozptylu. Výsledný obraz se šumem je poté přičten k danému obrazu.

6.1.3.9 Vlastní filtr

Tato metoda se skládá ze dvou funkcí, jedna, která vytvoří filtr a uloží jej do vnitřní proměnné objektu, to je možné udělat před načtení obrázku. Tato funkce vstupní data otáčí aby došlo ke konvoluci. Druhá funkce tyto data využije a použije funkci z OpenCV pro aplikování filtru, tato funkce dělá korelaci, proto je filtr v předchozí funkci otáčen.

6.1.3.10 Zkreslení objektivu

Zkreslení je vypočteno vytvořením odpovídající matic kamer a dodáním daných parametrů. To může být vytvořeno dvěma způsoby.

První způsob obsahuje dvě funkce. V první se definují kamery a koeficienty zkreslení z těchto parametrů jsou pak vygenerovány mapy, které jsou uloženy jako vnitřní proměnné objektu. Poté je využita funkce, která použije dané mapy k úpravě obrazu. Nevýhodou toho to způsobuje potřeba stejné velikosti obrázků.

Druhá funkce přeskakuje krok generování map, upravuje přímo dodaný obrázek pomocí funkce z OpenCV.

6.1.3.11 Vodní kapky

Efekt vodních kapek nebyl implementován původní navrženou metodou, podle skutečných fotografií. Pořízené fotografie, byly velmi odlišné a záleželo na velikosti kapek. Pokud byly kapky velmi malé obraz by spíše rozmazán. Když byly kapky velké, velká část obrazu v nich byla zmenšena a otočena vzhůru nohama.

Implementace proběhla pomocí vytvořených průhledných textur. Textur je celkem 6 ty jsou během funkce otočeny aby vniklo více možností, celkem možností je 36. Textury byly vytvořeny pomocí návodu v programu Gimp. [28]

Jednotlivé kanály textury jsou násobeny kanálem alfa, a poté přičteny k jednotlivým kanálům obrazu.



Obrázek 11 Vodní kapky – zaostřeno na kapky a velké kapky



Obrázek 12 Vodní kapky – bez kapek a malé kapky

7 UŽIVATELSKÉ ROZHRAŇÍ

Tato kapitola se bude věnovat uživatelskému rozhraní jeho implementaci a využití třídy, která implementuje jednotlivé metody. Uživatelské prostředí je vytvořeno v jazyce C++ za pomoci knihovny Qt. K vypracování a návrhu funkcí byla použita dokumentace knihovny Qt. [29]

Uživatelské rozhraní obsahuje čtyři třídy. Třída *MainWindow*, která zobrazuje hlavní okno a obsahuje funkci se smyčkou, která spouští dávkové úpravy obrázků. Třída *methodWidget* vytváří uživatelské rozhraní pro zadávání jednotlivých metod a náhled. Třída *CM2frame* vytváří část rozhraní pro výběr metod, které využívají vyhledávací tabulky. Třída *UndefPixFrame* vytváří rozhraní pro výběr typu nedefinovaný pixelů.

Načtení a uložení souborů je podporováno v těchto formátech:

- jpg, jpeg, jpe
- png
- bmp, dip
- tiff, tif

7.1 Třída *CM2frame*

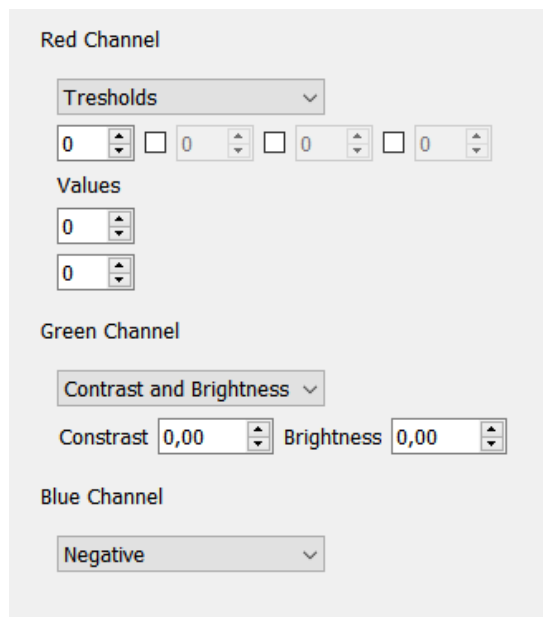
Tato třída vytváří rámeček pro výběr a zadání parametrů pro vytvoření vyhledávací tabulky. Výběrové možnosti jsou prahování, změna kontrastu a jasu, žádná změna a negativ. Možnost žádná změna, je možno schovat z výběru pomocí veřejného slotu.

Prahování tvoří 4 prvky *QspinBox*, ty určují dané prahy. Všechny jsou viditelné, první je přístupný a všechny další je nutné zpřístupnit zaškrtnutím sousedního *QcheckBox*. Pod těmi to prvky se nachází další *QspinBox* pro zadávání výše hodnot mezi prahy, při přidání prahu přibude vždy další prvek pro zadání hodnoty.

Změna kontrastu a jasu je tvořena dvěma *QdoubleSpinBox*, pro zadávání desetinných čísel.

Dané možnosti si přepínají mezi rozhraními schováním možností ostatních a zviditelnění svých prvků. Pro lepší manipulaci má každá možnost své prvky v rámečcích.

Předávání parametrů je zajištěno veřejnou funkcí, která předává parametry stejným způsobem jako metoda třídy *AugMethods* pro vytvořené vyhledávací tabulky.



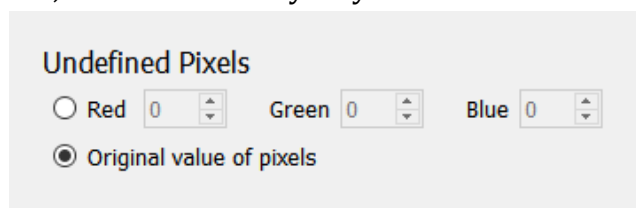
Obrázek 13 Objekty třídy CM2frame

7.2 Třída *UndefPixFrame*

V této třídě nalezneme prvky pro zadání parametrů nedefinovaných pixelů. Obsahuje dvě výběrové možnosti, jednu pro originální hodnotu pixelů a druhou pro výběr libovolné barvy podle RGB. Při překliknutí na možnost originální hodnoty jsou prvky pro zadávání barvy znepřístupněny.

Třída umožňuje při vytvoření její instance schovat možnost originální hodnoty pixelů, to bylo implementováno pro metodu oříznutí.

Vracení parametrů zajišťuje veřejná metoda, která zapisuje do dat hodnotu barev a zároveň vrací, která možnost byla vybrána.



Obrázek 14 Třída UndefPixFrame

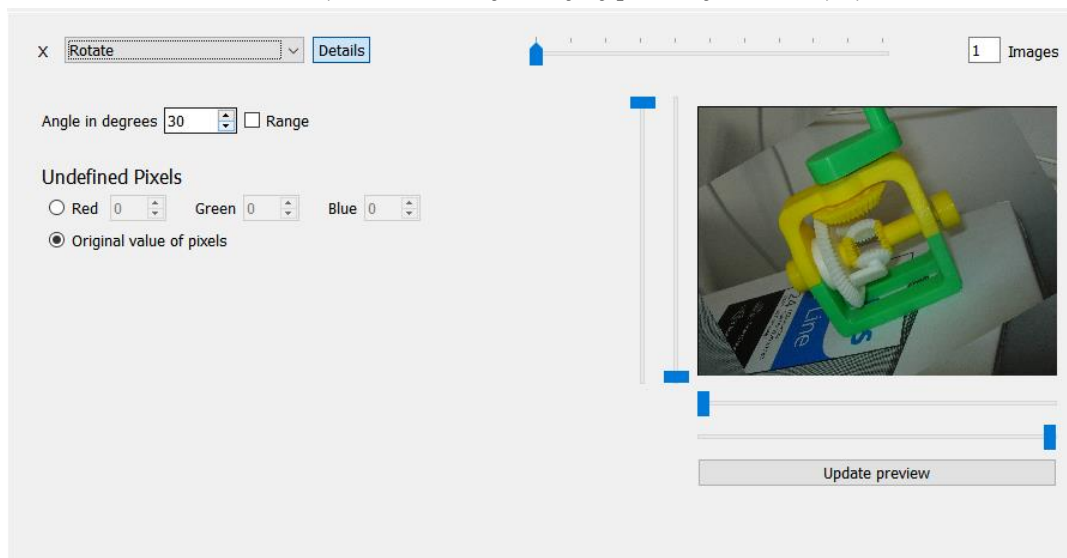
7.3 Třída *methodWidget*

Tato třída vytváří prostředí pro zadávání parametrů ke všem metodám. Každá metoda nebo jejich skupina má vlastní stránku ve *QStackedWidget*, který umožňuje přepínat stránky mezi sebou.

Ve vrchní části se nachází tlačítko *X*, které má zničit objekt, toho je docíleno vysláním signálu prvku ve kterém byla instance vytvořena. Další objektem v horní části je nabídka jednotlivých metod, výběrem se mění stránky s metodami. Tlačítko

Details, umožňuje zobrazit a nebo schovat oblast pro zadávání parametrů metod a náhled obrázku. Posuvníkem se určuje počet nově vytvořených obrázků, přesné číslo se nachází vedle v políčku. Změna počtu obrázků vysílá signál o její změně.

V souboru se třídou je obsažen výčtový typ, který obsahuje jednotlivé metody.



Obrázek 15 Třída `methodWidget`

7.3.1 Náhledový obrázek

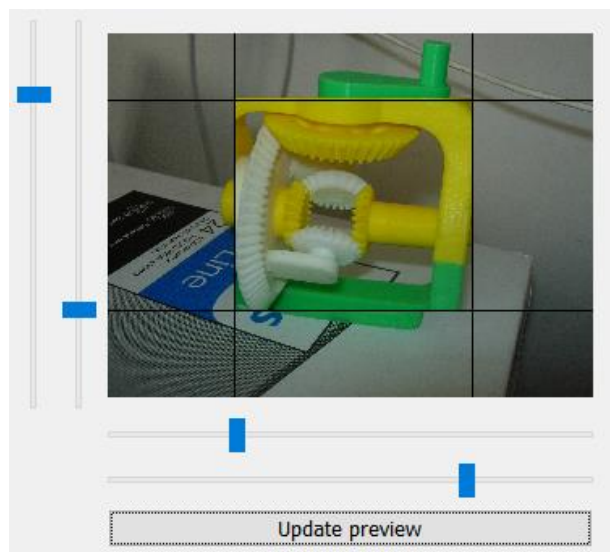
Jednou z vnitřních proměnných třídy *methodWidget* je objekt třídy *AugMethods* s názvem *image*. Tento objekt je statický a je inicializován statickou veřejnou funkcí. Tato funkce je poté spouštěna v *MainWindow* při načtení obrázků a je při ní nahrán první obrázek do daného objektu. Objekt je statický z důvodu neexistence žádné instance dané třídy při načtení obrázků.

Při stisku tlačítka, které obnoví náhled, dojde k spuštění funkce, která spustí pomocí výběru danou metodu na objekt *image*. Poté je obrázek převeden do správného typu a zobrazen v *Qlabel*. Obrázek se obnoví pouze při stisku tlačítka. Před prvním obnovení je zobrazen původní.

7.3.1.1 Lokalizace

Výběr oblasti, ve kterém bude vybraná metoda působit je umožněn v náhledu obrázku, okolo nacházejícími se posuvníky. Posuvníky vykreslují ve stejné úrovni jako jsou vůči obrázku, černou čáru pro znázornění vybírané oblasti. Při změně polohy posuvníků jsou tyto čáry překresleny. Při překreslení je nutné přepočítávat souřadnice.

Získání parametrů je věnována veřejná metoda, která vyčte hodnoty z posuvníků a převede je na správné souřadnice.



Obrázek 16 Náhledový obrázek s lokalizací

7.3.2 Jednotlivé metody

Funkce jednotlivých metod jsou velmi podobné. Každá metoda má svoji vlastní veřejnou metodu, která vrací zadané parametry. Pro vrácení, která funkce byla vybrána je vytvořena další metoda, ta pro větší přehlednost vrací výčtový typ.

Pokud je na stránce dané metody výběrové tlačítko `QRadioButton`, je ošetřeno, aby do prvků ostatních možností, než je vybraná, nebylo možno zapisovat. Jestliže se u dané metody nachází zaškrtnutí možnosti pro výběru z rozmezí dvou hodnot, je při vybrání této možnosti zobrazeno další políčko pro vložení hodnoty. Hodnota je pak z daného intervalu vybrána náhodně, kdy první hodnota do intervalu patří a poslední ne.

U metod, kde vznikají nedefinované pixely, je zobrazena třída `UndefPixFrame`. Pro zjištění parametrů je vytvořena veřejná metoda, která vybere podle zvolené metody správný objekt této třídy a vrátí jeho parametry.

7.3.2.1 Zrcadlení

Zrcadlení nabízí tři možnosti jak obrázek změnit, ty jsou přetočení kolem osy X, Y nebo obou zároveň.

7.3.2.2 Rotace

Rotace nabízí možnost výběru otočení obrázku o daný úhel.

7.3.2.3 Posunutí

U posunutí je na výběr možnost posunutí v ose X a v ose Y a to buď v procentech nebo v pixelech.

7.3.2.4 Škálování

Škálování umožňuje zvětšení nebo zmenšení obrazu v procentech v ose X a Y.

7.3.2.5 Oříznutí

Oříznutí nabízí výběr oblasti v daných osách. Tato oblast bude zachována.

7.3.2.6 Zkosení

U zkosení je možné nastavit velikost zkosení v dané ose.

7.3.2.7 Metody pro úpravu barev

Tato metoda obsahuje několik možností, jak obraz upravit. Možnost nastavení daného kanálu na konstantní hodnotu, pro barevný prostor RGB a HSV.

Druhá možnost umožňuje přičtení a odečtení hodnot od všech kanálů, pro barevné prostory RGB a HSV. Možnost Hue je omezena na 180, protože taková je velikost kanálu Hue v OpenCV.

Třetí možnost převádí obraz do šedotónového. Poslední možnost vydělí počet barev pro každý kanál zadaným číslem.

7.3.2.8 Prahování a úpravy jasu

Tato umožňuje vybrat buď pro každý jeden kanál nebo všechny kanály dohromady metodu úpravy. Výběr metod a jejich parametrů je uskutečněn pomocí objektů třídy *CM2frame*.

7.3.2.9 Šum

Možné je vybírat ze dvou šumů. Pro impulsní šum je možné nastavit procentuálně kolik pixelů bude ovlivněno a jaké barvy budou.

U Gaussova šumu je možné změnit střední hodnotu a rozptyl.

7.3.2.10 Filtry

Z filtrů je možné vybrat 3 filtry pro vyhlazování. U každého je možné zvolit jeho velikost, pro Gaussovo rozmazání je možné zadat směrodatnou odchylku. Pro vytvoření vlastní ho filtru je zahrnuta možnost vytvoření filtru o velikosti 5x5 s možností vynásobení všech hodnot konstantou.

7.3.2.11 Zkreslení objektivu

Zkreslení objektivu umožňuje výběr parametrů zkreslení podle rovnic 5.1, 5.2, 5.3 a 5.4.

7.4 Třída *MainWindow*

Instance třídy *MainWindow* je spouštěna v *main.cpp* a je na ní postaven zbytek celého programu. Probíhá zde dávkové spuštění úpravy obrázků a zobrazuje hlavní okno programu.

7.4.1 Načtení souborů

Před načtením obrázku je se všemi prvky znemožněna interakce. Pro načtení obrázků je možno použít dvou metod. První otevírá složku a druhá umožňuje výběr daných obrázků. Pokud je výběr neplatný vyskočí okno s upozorněním. Jména načtených souborů, splňující podmínky správného typu, jsou uloženy ve vnitřní proměnné a je inicializována statická proměnná pro náhled třídy *methodWidget*. Po načtení jsou další prvky objektů zpřístupněny pro interakci.

7.4.2 Zobrazení metod

Přidáním metody je vytvořen nový objekt třídy *methodWidget*. Který je zobrazen v oblasti, která umožňuje scrollování. Množství nových metod není omezeno. Metody nelze mezi sebou přesouvat, každá nová lišta s možností výběru metody je dána na konec seznamu. Pokud je objekt odalokován, na základě signálu, který vyslal, je odstraněn pointer z vnitřních proměnných a zavolána funkce pro bezpečné odalokování.

7.4.3 Uložení

Ve spodní části se nachází nabídka pro výběr formátu k uložení. Cesta pro uložení může být napsána buď ručně nebo vybrána pomocí vyskakovacího okna. Cesta pro uložení musí existovat, pokud neexistuje obrázek nebude uložen. Do lišty jde za znak „/“ vepsat počáteční název souboru, za něj je pak připojeno pořadové číslo. Při ukládání se zobrazuje název posledního uloženého souboru ve spodní liště okna.

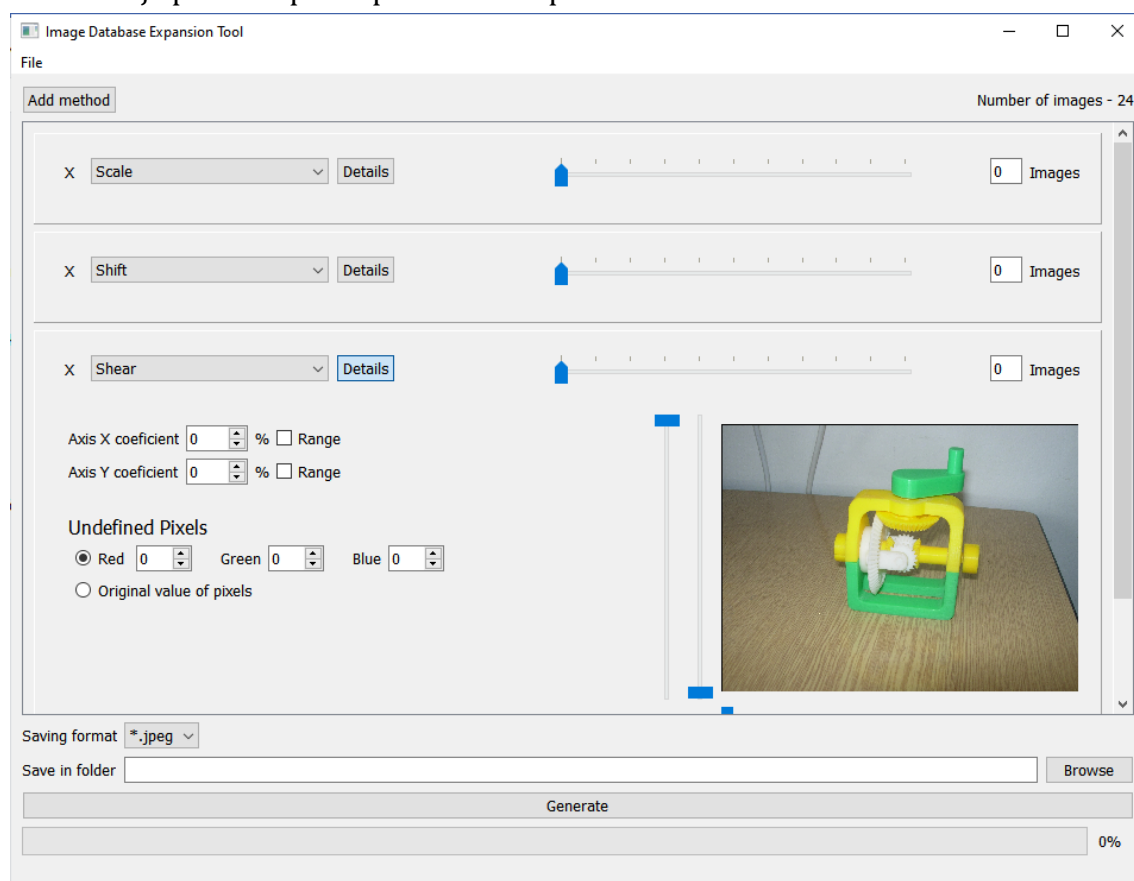
7.4.4 Spuštění generování obrázků

Po spuštění generování nových obrázků, je znemožněno interagovat s prvky programu. Postupně běží smyčka ve které jsou procházeny jednotlivé objekty třídy *methodWidget*. Pro každý z těchto objektů je spuštěna funkce, která provádí spuštění dané metody.

Během této funkce je inicializována lišta, která ukazuje kolik obrázků z dané metody bylo již vytvořeno. Předání parametrů lokalizace probíhá před rozdělením operací a jejich smyčkami. Po výběru dané operace jsou získány jejich parametry. Poté probíhá smyčka ve které jsou obrázky upraveny, pokud byla vybrána rozmezí

hodnot je vybrán náhodný parametr. Smyčka je pro každou operaci zvlášť z důvodu jednoduššího získání proměnných jako je vyhledávací tabulka a konvoluční filtry. Ve smyčce probíhá načtení obrázku, jeho úprava a následné uložení. Tento přístup znemožňuje kombinovat metody dohromady na jednom obrázku, ale výběr dané správné operace se vykoná pouze jednou.

Po dokončení generování nových obrázků vyskočí okno s počtem úspěšných uložení a je povolen přístup k vnitřním prvkům.



Obrázek 17 Okno programu

8 ZHODNOCENÍ A VÝSLEDKY TESTŮ

V této kapitole bude popsán výsledek testování. Popsány omezující podmínky programu a jeho celkové zhodnocení.

8.1 Výsledky testů

Při testu, který zkoušel vytvořit velké množství obrázků, došlo k problému, že okno aplikace nebylo dostatečně často updatováno. Tento problém byl vyřešen přidáním volání update okna do smyčky, ve které se tvoří obrázku. Poté byly vyzkoušeny všechny metody s maximální počtem obrázků, tedy 999, nebyly nalezeny žádné chyby.

V dalším testu byly odzkoušeny všechny možné parametry a jejich správná činnost. Při zadávání hodnot byly zkoušeny kladné a záporné hodnoty. Při vytváření vyhledávací tabulky, byly testovány kanály dohromady a jednotlivě, zdali vytváří stejné obrázky a jestli je výběr parametrů kanálu nezávislý na ostatních kanálech. Test prošel bez větších chyb a drobné byly opraveny.

Při ověření výběru obrázků, se testoval, náhodný výběr obrázků při větším množství než počet nahraných obrázků. Dále jestli jsou vybrány všechny obrázky při stejném množství generovaných obrázků jako metoda alternuje. Pokud obrázků bylo v metodě méně, neměl by se obrázek opakovat v generovaných. Test prošel bez chyb.

Při ověření správné podpory typů obrázků, byly nejprve vygenerovány nové obrázky ve všech typech, ověřena jejich správnost a poté načteny a z nich vygenerován jeden typ. Test prošel bez obtíží.

8.2 Zhodnocení a omezující podmínky

V této kapitole bude popsán stav programu a jeho nedostatky a jejich možné řešení, které nebyli udělány pro svoji časovou náročnost začlenění řešení, rozdílný přístup k programu nebo protože jejich změna představovala přepsání již značnou část programu.

Program je plně funkční, při běhu by neměla být žádná chyba. Umožňuje výběr několika metod s různými parametry, některé nabízí výběr rozmezí, které umožňuje náhodné vybrání parametru. Program je schopen generovat velké množství obrázků.

Programu chybí přesnější výběr parametrů, pro většinu možností nabízí výběr pro procenta v celých číslech. Tento výběr mohl být změněn na výběr do dvou desetinných čísel. Tato změna nebyla provedena z důvodu mnoha provázaných

funkcí mezi jednotlivými prvky, přetypování těchto funkcí a kontrola správnosti by zabrala mnoho času.

Mnohem více parametrů mohlo být provedeno náhodným výběrem, který je například vidět u funkce rotace. U funkcí, které používají vyhledávací tabulku, vytváří vlastní filtr a u funkce zkreslení objektivu by použití náhodných parametrů značně snížilo rychlost vytvoření nových obrázků. Každá z nich si generuje pomocnou proměnnou pro všechny obrázky stejnou, pokud by byli použity náhodné parametry musela by se generovat pro každý obrázek nová.

Další možná změna a nedostatek programu je nemožnost kombinovat metody úprav na jednom obrázku. Pro vytvoření této změny by muselo být změněno uživatelské prostředí a způsob vytváření obrázků ve smyčce, zde by nastala zase obtíž s vyhledávacími tabulkami a jinými podobně fungujícími proměnnými.

Pro náhled vzniká chyba při jeho aktualizaci v jedné metodě a při pohnutí posuvníky pro výběr oblasti lokalizace. Při aktualizaci obrázku v dané metodě je náhled již v pořádku To je zapříčiněno použitím statické proměnné, která uchovává obrázek. Možné řešení by bylo nepoužít statickou proměnnou, kdy by o danou proměnnou byly objekty jednotlivých metod větší, tato možnost by pravděpodobně byla nejlepší. Druhou možností je vykonat danou metodu pro daný obrázek. Nevýhodou této možnosti by bylo zdlouhavé aktualizování obrázku, pokaždé když by se pohnulo posuvníkem.

Pro lokalizaci nejsou použity posuvníky s dvěma jezdcí, protože knihovna Qt, je neobsahuje, musely by být vytvořeny v rámci programu. To je také jeden z důvodů, proč nejsou u žádného výběru jednotlivých parametrů použity posuvníky.

Při ukládání souborů je v názvu obrázku číslovka vždy minimálně o 5-ti místech, velikost této číslovky by mohla být vybírána podle celkového počtu obrázků. To by muselo být řešeno složitější prací s textovými řetězci.

Velkou nevýhodou mého programu je začlenění smyčky pro generování obrázků do kódu hlavního okna, i přesto, že tato možnost zjednodušuje aktualizování hlavního okna při generování obrázků. Tento přístup by umožňoval úplné oddělení části uživatelského rozhraní a části, která generuje a upravuje obrázky. Tento způsob by umožnil použití třídy pro generování obrázků samostatně. Aktualizace okna při běhu by poté byla lépe řešena použitím dvou paralelních vláken.

Metoda kapek mohla generovat kapky způsobem, jakým byly vytvořeny textury kapek, i přesto, že by poté byla metoda časově náročná. Výhodou by bylo vytvoření náhodných kapek s náhodnou velikostí, odpovídající velikostně vstupnímu obrazu. Momentální nedostatky této metody jsou příliš malá velikost kapek při velkých obrázcích. A malý počet různých textur, tento nedostatek by šel částečně obejít vygenerováním více textur.

9 ZÁVĚR

Tato práce se zabývala vytvořením nástroje pro rozšíření obrazových databází. To zahrnovalo návrh jednotlivých metod pro rozšíření obrazových databází, jejich implementaci a vytvoření uživatelského rozhraní pro tyto metody s dávkovým spuštěním.

První dvě části se věnovaly zpracování obrazu a existujícím nástrojům a přístupům k dané problematice augmentace obrazu. Část věnovaná zpracování obrazu poskytuje teoretický základ pro další zbytek práce. Další část se věnovala existujícím přístupům a nástrojům. Nejčastěji se datová augmentace řeší pro neuronové sítě, tomu také odpovídaly nástroje a metody v této části popsané.

Při návrhu metod bylo využito teoretického základu z předchozích částí. Většina navržených metod využívá jednoduchých úprav obrázků, nejen že takové metody jsou rychlé, ale často i velmi efektivní. Návrh metody, která přidává vodní kapky, nebyl použit, místo něj byl využit jiný způsob, který je popsán v další kapitole.

Testy byly zvoleny, aby vyzkoušely jednotlivé části programu, tedy zkoušely všechny specifické parametry, zdali je program schopen generovat velká množství a zdali je správně implementován náhodný výběr.

Při implementaci funkcí byla udělána již zmíněná změna. Implementace byla zařízena pomocí jedné třídy, která zastřešovala všechny metody a umožňovala oddělit proměnné obrazu od programu. Při implementaci byla snaha o nepříliš náročné metody.

Uživatelské rozhraní tvoří největší část programu. Zajišťuje nejen zadávání parametrů, ale i dávkové spuštění úprav obrázků. Každá metoda má vytvořenou vlastní část rozhraní. Jednotlivých metod lze použít na jednu načtenou databázi neomezené množství. Při zapnutí generování obrázků je spuštěna hlavní část programu, která zajišťuje vytvoření nových obrázků.

V poslední části se práce věnuje testům. Testy dopadly na poslední verzi programu bez chyb. Dále se část věnuje nedostatkům programu a jejich možným řešením a odůvodnění, proč tato řešení nebyla použita.

10 LITERATURA

- [1] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 2. přeprac. vyd. Praha: ČVUT, 2007, 255 s. : il. ISBN 978-80-01-03110-0.
- [2] AL-NAJJAR, Yusra. Histogram equalization for the original and the enhanced images. In: *Research Gate* [online]. 2018 [cit. 2020-01-04]. Dostupné z: <https://www.researchgate.net/>
- [3] ARDENSON, Richard a Peter KROGH. Color space and color profiles. *DpBestflow.org* [online]. 2015 [cit. 2020-01-04]. Dostupné z: <https://www.dpbestflow.org/color/color-space-and-color-profiles>
- [4] GONZALEZ, Rafael a Richard WOODS. *Digital image processing*. 3rd ed. Upper Saddle River: Pearson ; Prentice Hall, 2008, xxii, 954 s. : il. ISBN 0-13-168728-X.
- [5] RGB color solid cube. In: *Wikimedia Commons* [online]. [cit. 2020-01-04]. Dostupné z: <https://commons.wikimedia.org/>
- [6] The HSV color model mapped to a cylinder. In: *Wikimedia Commons* [online]. [cit. 2020-01-04]. Dostupné z: https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder.png
- [7] SHARMA, Tejpal. Various Types of Image Noise and De-noising Algorithm. *International Journal of Modern Education and Computer Science* [online]. Hong Kong: Modern Education and Computer Science Press, 2017, **9**(5), 50 [cit. 2020-01-04]. DOI: 10.5815/ijmecs.2017.05.07. ISSN 20750161. Dostupné z: <http://search.proquest.com/docview/1984781306/>
- [8] SINHA, Utkarsh. The result of convolution with the laplacian operator. In: *AI Shack* [online]. [cit. 2020-01-05]. Dostupné z: <http://www.aishack.in/tutorials/sobel-laplacian-edge-detectors/>
- [9] GOODFELLOW, Ian, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDE-FARLEY, Sherjil OZAIR, Aaron COURVILLE a Yoshua BENGIO. Generative Adversarial Nets. In: *Proceedings of the International Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2014, s. 2672-2680.
- [10] ANTREAS, Antoniou, Storkey AMOS a Edwards HARRISON. Data Augmentation Generative Adversarial Networks. *ArXiv* [online]. 2017, (171104340) [cit. 2020-05-31].
- [11] BUSLAEV, Alexander, Vladimir IGLOVIKOV, Eugene KHVEDCHENYA, Alex PARINOV, Mikhail DRUZHININ a Alexandr KALININ. *Albumentations: Fast and*

- Flexible Image Augmentations* [online]. 2020 [cit. 2020-05-31]. DOI: 10.3390/info11020125. Dostupné z: <https://www.mdpi.com/2078-2489/11/2/125>
- [12] JUNG, Alexander, Kentaro WADA, Jon CRALL, Satoshi TANAKA, Jake GRAVING, Christoph REINDERS a Sarthak YADAV. *Imgaug* [online]. [cit. 2020-05-31]. Dostupné z: <https://github.com/aleju/imgaug>
- [13] XIE, Qizhe, Zihang DAI, Eduard HOVY, Minh-Thang LUONG a Quoc LE. Unsupervised Data Augmentation for Consistency Training. *ArXiv: preprint arXiv*. 2019, (190412848).
- [14] *Data Augmentation For Object Detection* [online]. [cit. 2020-05-31]. Dostupné z: <https://github.com/Paperspace/DataAugmentationForObjectDetection>
- [15] KANG, Guoliang, Xuanyi DONG, Liang ZHENG a Yi YANG. PatchShuffle Regularization. *ArXiv.org* [online]. Ithaca: Cornell University Library, arXiv.org, 2017 [cit. 2020-01-06]. Dostupné z: <http://search.proquest.com/docview/2075652691/>
- [16] INOUE, Hiroshi. Data Augmentation by Pairing Samples for Images Classification. *ArXiv.org* [online]. Ithaca: Cornell University Library, arXiv.org, 2018 [cit. 2020-01-06]. Dostupné z: <http://search.proquest.com/docview/2071988663/>
- [17] SUMMERS, Cecilia a Michael DINNEEN. Improved Mixed-Example Data Augmentation. *ArXiv.org* [online]. Ithaca: Cornell University Library, arXiv.org, 2019 [cit. 2020-01-06]. Dostupné z: <http://search.proquest.com/docview/2073843532/>
- [18] MIKOLAJCZYK, Agnieszka a Michal GROCHOWSKI. Data augmentation for improving deep learning in image classification problem. In: *2018 International Interdisciplinary PhD Workshop (IIPHDW)* [online]. IEEE, 2018, s. 117-122 [cit. 2020-01-06]. DOI: 10.1109/IIPHDW.2018.8388338.
- [19] ZHONG, Zhun, Liang ZHENG, Guoliang KANG, Shaozi LI a Yi YANG. Random Erasing Data Augmentation. *ArXiv.org* [online]. Ithaca: Cornell University Library, arXiv.org, 2017 [cit. 2020-01-06]. Dostupné z: <http://search.proquest.com/docview/2076552947/>
- [20] SHORTEN, Connor a Taghi KHOSHGOFTAAR. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* [online]. Cham: Springer International Publishing, 2019, 6(1), 1-48 [cit. 2019-12-29]. DOI: 10.1186/s40537-019-0197-0. Dostupné z: <https://journalofbigdata.springeropen.com/>

- [21] ZHANG, Aston, Zachary LIPTON a Alexander SMOLA. 13.1. Image Augmentation. *Dive into Deep Learning* [online]. [cit. 2019-12-29]. Dostupné z: <https://d2l.ai/>
- [22] UHER, Václav. *Trénovatelné metody pro automatické zpracování biomedicínských obrazů* [online]. Brno, 2018 [cit. 2019-12-29]. Dostupné z: <http://hdl.handle.net/11012/137277>. Disertační práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Radim Burget.
- [23] LI, Hongwei. An example of data augmentation result.: From left to right: the original axial slice, slice after rotation, slice after shear mapping and slice after scaling. Source publication. In: *Research Gate* [online]. [cit. 2020-01-06]. Dostupné z: <https://www.researchgate.net/>
- [24] Jasové transformace. In: *Multimediální interaktivní didaktický systém* [online]. 2010 [cit. 2020-01-06]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php
- [25] Color Reduction Basics. In: *Graphics Mill* [online]. [cit. 2020-01-06]. Dostupné z: <https://www.graphicsmill.com/docs/gm/color-reduction-basics.htm>
- [26] HORÁK, Karel. Geometrické transformace. *MIDAS: Zpracování vícerozměrných signálů* [online]. 2010 [cit. 2020-06-02]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php
- [27] *OpenCV: OpenCV 4.2.0* [online]. 2019 [cit. 2020-06-02]. Dostupné z: <https://docs.opencv.org/4.2.0/>
- [28] Tutorial: How to generate realistic mass waterdrops on a surface. *GIMPUSERS.com* [online]. [cit. 2020-05-26]. Dostupné z: <https://www.gimpusers.com/tutorials/waterdrops-on-a-surface>
- [29] *Qt: 5.12* [online]. 2020 [cit. 2020-06-02]. Dostupné z: <https://doc.qt.io/qt-5.12/index.html>